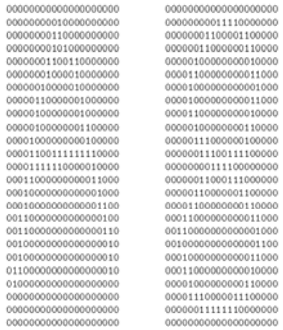
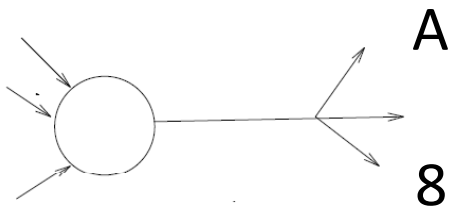
4º CAPÍTULO

Conceitos de Reconhecimento de Padrões





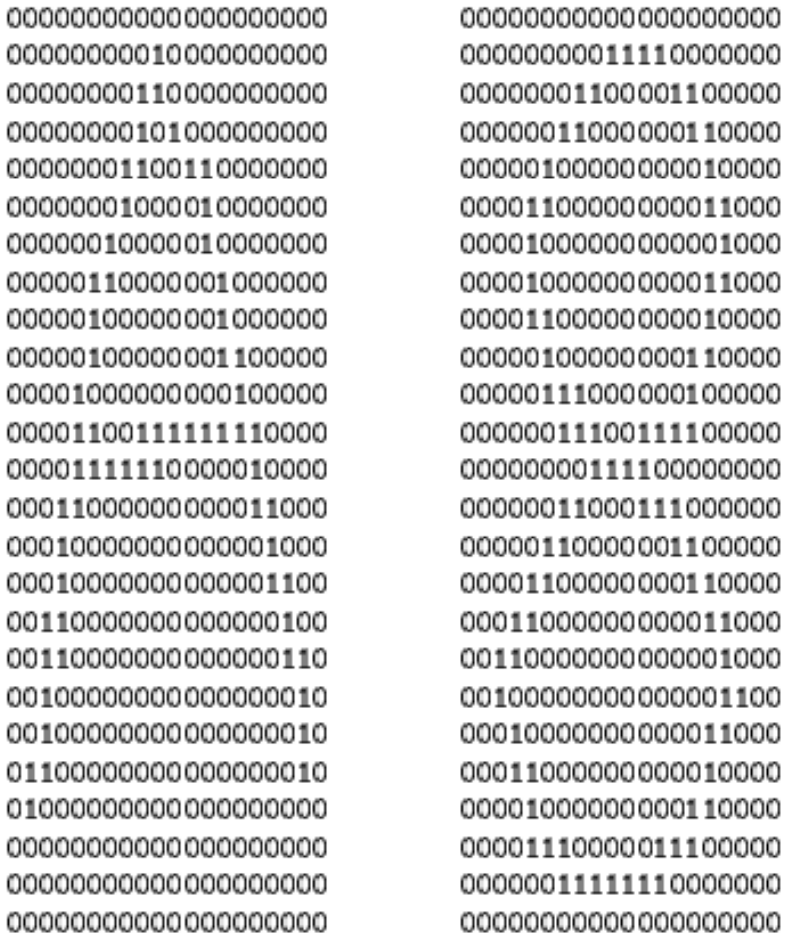
Prof. Arnaldo Abrantes / anotado por Prof. Nuno Pinho da Silva

Reconhecimento de padrões

- ***Definição:*** ao processo de atribuir uma etiqueta (classe) à instância de objecto observado designa-se por **classificação**
  - Porquê a necessidade da classe de rejeição?
  - Exemplo: reconhecimento de caracteres, classificação de frutos

E se alguém inventar um novo caracter, ou desenhar um 'a', e esta informação for fornecida ao sistema de classificação?

- ***Definição:*** ao processo de fazer corresponder (ou não) uma instância de um objecto com um protótipo específico designa-se por **verficação**
  - Exemplo: verificação de identidade num ATM



Imagens Binárias: caracteres 'A' e '8'

Reconhecimento: identificação de algo ou alguém já conhecido (infopedia.pt)

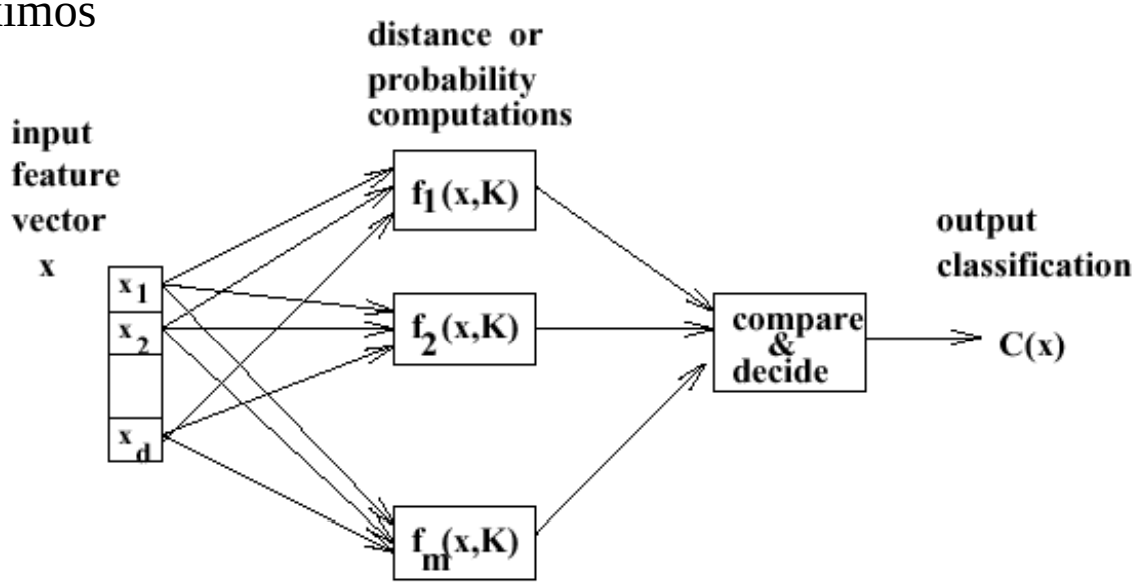
Reconhecimento – depreende conhecimento prévio sobre os objectos que se pretendem classificar – classificação supervisionada

Modelo de sistema de classificação

- Componentes dum sistema de classificação
  - Classes $C_1, C_2, \cdots, C_{m-1}$
    - classe de rejeição $C_m = C_r$
  - Sensores
  - Extracção de características (*features*)
  - Classificador
    - assinatura mais próxima
    - k-vizinhos mais próximos
    - MAP
    - rede neuronal
    - árvore de decisão

A extracção de características torna ,normalmente, o processo de classificação independente do problema de recolha de informação, ou seja, pode-se aplicar o mesmo tipo de classificador (ajustado ao problema) a várias aplicações de reconhecimento

Assumindo que a imagens de input são do mesmo tamanho, que características poderiam ser úteis para classificar caracteres em imagens binárias?



Avaliação do sistema

- Definições:
  - o classificador comete um ***erro de classificação*** sempre que classifica um objecto na classe C_i e a verdadeira classe é C_j $(i \neq j; C_i \neq C_r)$
  - ***taxa experimental de erros*** é igual ao número de erros de classificação, cometidos nos dados de teste independentes, a dividir pelo número de testes efectuadas
  - ***taxa experimental de rejeições*** é igual ao número de rejeições feitas nos dados de teste independentes a dividir pelo número de testes efectuados
  - designa-se por ***dados de teste independentes*** a um conjunto de amostras de classe conhecida, incluindo objectos da classe rejeição, que não foram usados no desenvolvimento do algoritmo de extracção de características nem no algoritmo de classificação (não foram usados no treino)

Desempenho é avaliado por ambas as taxas.

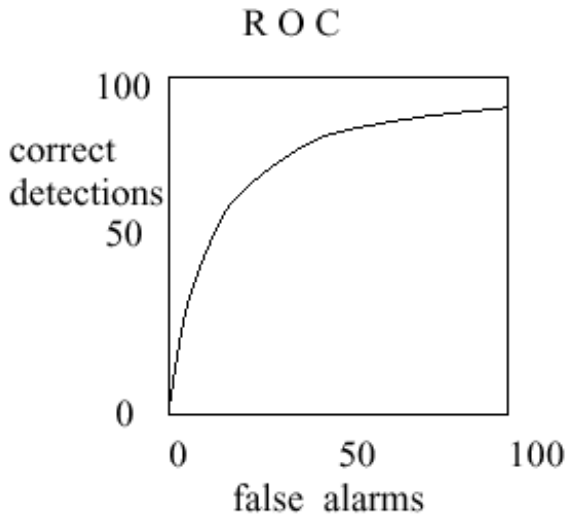
Um sistema que classifique todos os itens na classe de rejeição não comete nenhum erro mas é inútil

Avaliação do sistema (cont.)

- Matriz de confusão

		class j output by the pattern recognition system										
		'0'	'1'	'2'	'3'	'4'	'5'	'6'	'7'	'8'	'9'	'R'
	'0'	97	0	0	0	0	0	1	0	0	1	1
	'1'	0	98	0	0	1	0	0	1	0	0	0
true	'2'	0	0	96	1	0	1	0	1	0	0	1
object	'3'	0	0	2	95	0	1	0	0	1	0	1
class	'4'	0	0	0	0	98	0	0	0	0	2	0
	'5'	0	0	0	1	0	97	0	0	0	0	2
i	'6'	1	0	0	0	0	1	98	0	0	0	0
	'7'	0	0	1	0	0	0	0	98	0	0	1
	'8'	0	0	0	1	0	0	1	0	96	1	1
	'9'	1	0	0	0	3	0	0	0	1	95	0

- Problemas envolvendo apenas 2 classes
  - Exemplos
    - objecto em bom estado / objecto em mau estado
    - objecto presente na imagem / objecto ausente
    - pessoa com doença D / pessoa sem doença D
  - Conceitos
    - Falsos alarmes (falso positivo) vs
      falhas de detecção (falso negativo)
    - Precisão vs rechamada



Avaliação do sistema (cont.)

		class j output by the pattern recognition system										
		'0'	'1'	'2'	'3'	'4'	'5'	'6'	'7'	'8'	'9'	'R'
	'0'	97	0	0	0	0	0	1	0	0	1	1
	'1'	0	98	0	0	1	0	0	1	0	0	0
true	'2'	0	0	96	1	0	1	0	1	0	0	1
object	'3'	0	0	2	95	0	1	0	0	1	0	1
class	'4'	0	0	0	0	98	0	0	0	0	2	0
	'5'	0	0	0	1	0	97	0	0	0	0	2
i	'6'	1	0	0	0	0	1	98	0	0	0	0
	'7'	0	0	1	0	0	0	0	98	0	0	1
	'8'	0	0	0	1	0	0	1	0	96	1	1
	'9'	1	0	0	0	3	0	0	0	1	95	0

$$\text{Precisão} = \frac{Verdadeiros\ Positivos}{Verdadeiros\ Positivos + Falsos\ Positivos}$$

$$\text{Rechamada} = \frac{Verdadeiros\ Positivos}{Verdadeiros\ Positivos + Falsos\ Negativos}$$

Para este classificador multi-classe:

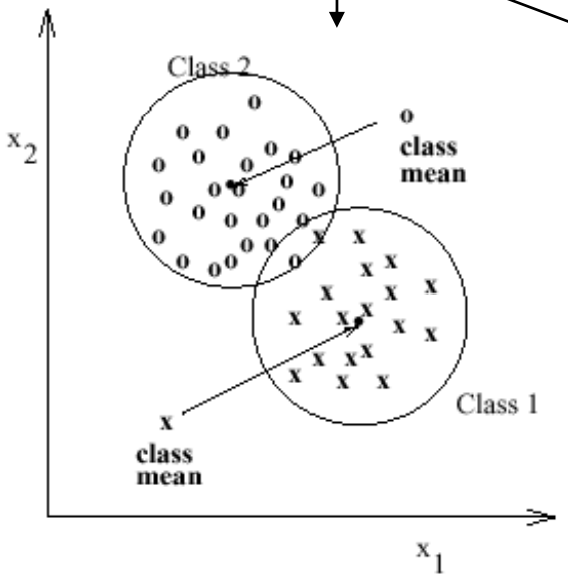
1. Calcule a taxa experimental de erro e a taxa experimental de rejeições
   2.1. Para cada classe
2. Calcule os falsos alarmes e as falhas de deteção
   2.1. Para cada classe
3. Calcule a precisão e a rechamada (precisão-rechamada)
   3.1. Para cada classe

Vectores de *features* e classificação

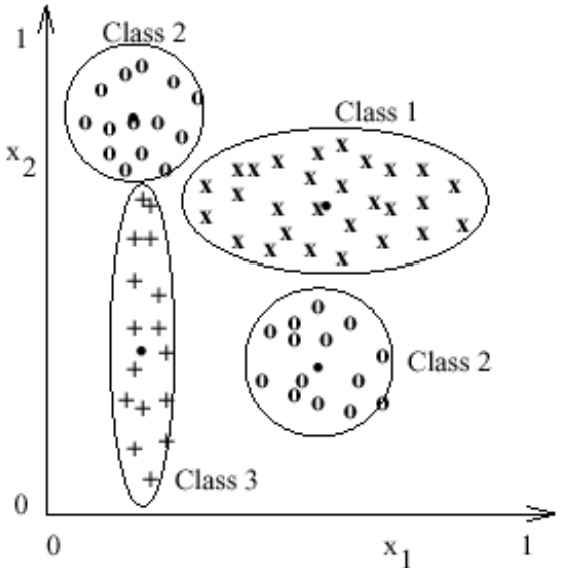
- Vectores de *features*
  - 8D
  - 2D

(class) character	area	height	width	number #holes	number #strokes	(cx,cy) center	best axis	least inertia
'A'	medium	high	3/4	1	3	1/2,2/3	90	medium
'B'	medium	high	3/4	2	1	1/3,1/2	90	large
'8'	medium	high	2/3	2	0	1/2,1/2	90	medium
'0'	medium	high	2/3	1	0	1/2,1/2	90	large
'1'	low	high	1/4	0	1	1/2,1/2	90	low
'W'	high	high	1	0	4	1/2,2/3	90	large
'X'	high	high	3/4	0	2	1/2,1/2	?	large
'*'	medium	low	1/2	0	0	1/2,1/2	?	large
'-'	low	low	2/3	0	1	1/2,1/2	0	low
'/'	low	high	2/3	0	1	1/2,1/2	60	low

Segundo momento dos píxeis em relação aos eixo de menor inérica



$$\|\mathbf{x}-\mathbf{x}_c\| = \sqrt{\sum_{i=1}^{d}(\mathbf{x}_1[i]-\mathbf{x}_2[i])^2}$$



$$\|\mathbf{x}-\mathbf{x}_c\| = \sqrt{\sum_{i=1}^{d}((\mathbf{x}_1[i]-\mathbf{x}_2[i])/\sigma_i)^2}$$

Classificadores mais simples:
- "if then else"
- Distância ao centróide

Classificador – k-vizinhos mais próximos

S is a set of n labeled class samples s_i where $s_i.\mathbf{x}$ is a feature vector and $s_i.c$ is its integer class label.
x is the unknown input feature vector to be classified.
A is an array capable of holding up to k samples in sorted order by distance d.
The value returned is a class label in the range **[1, m]**

```
procedure K_Nearest_Neighbors(x, S)
{
make A empty;
for all samples s_i in S
{
   d = Euclidean distance between s_i and x;
   if A has less than k elements then insert (d, s_i) into A;
   else if d is less than max A
      then {
              remove the max from A;
              insert (d, s_i) in A;
           }
} ;
assert A has k samples from S closest to x;
if a majority of the labels s_i.c from A are class c_0
   then classify x into class c_0;
   else classify x into the reject class;
return(class_of_x);
}
```

kNN

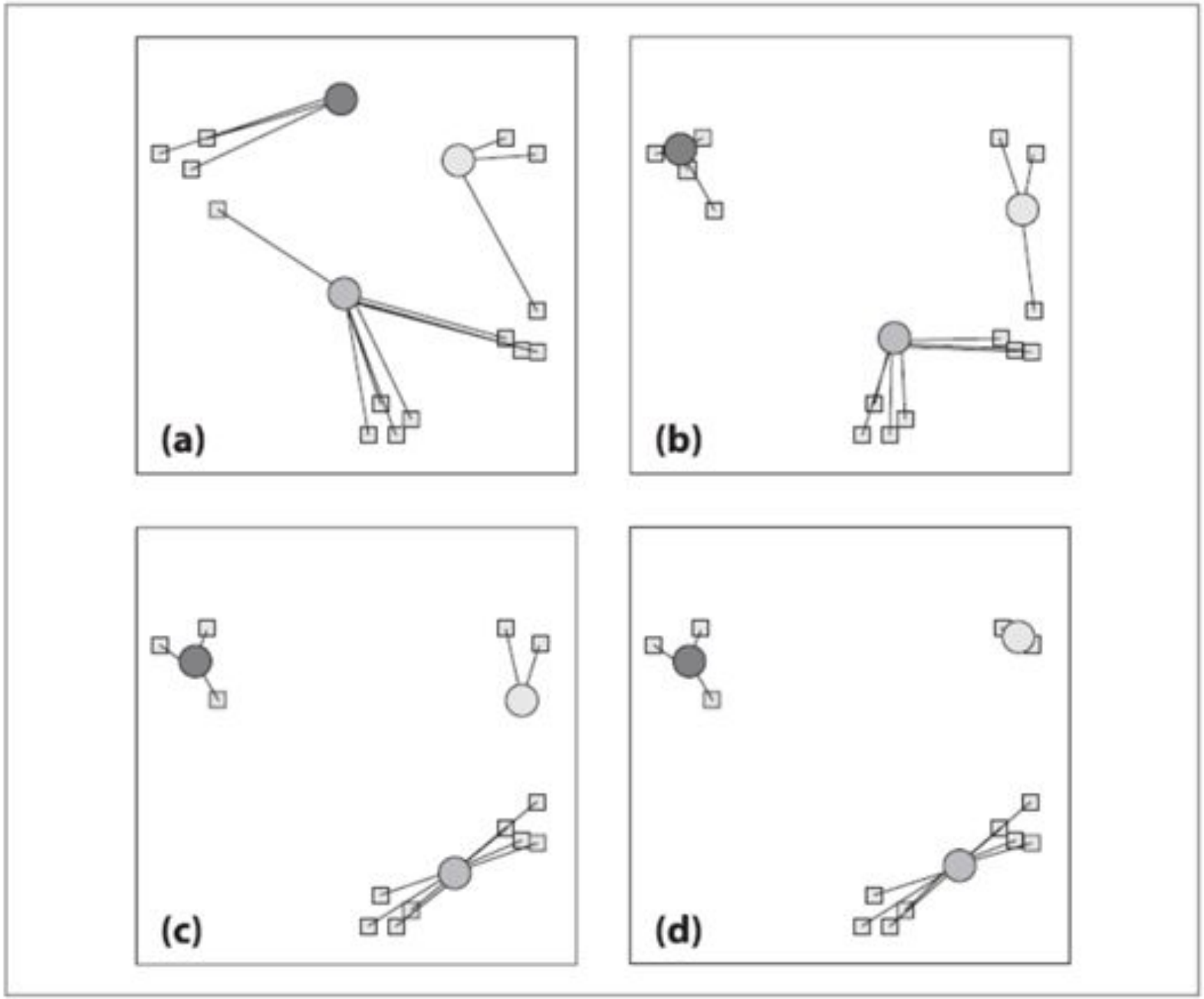


Figure 13-5. K-means in action for two iterations: (a) cluster centers are placed randomly and each data point is then assigned to its nearest cluster center; (b) cluster centers are moved to the centroid of their points; (c) data points are again assigned to their nearest cluster centers; (d) cluster centers are again moved to the centroid of their points

source: Learning OpenCV

Trabalho Prático

O TP1 está publicado no moodle.

Algumas notas:

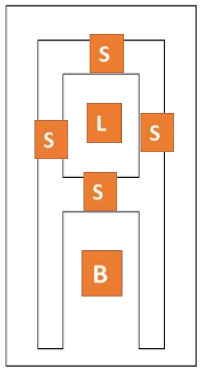
1. Sugestão de organização do relatório:
   - Capa;
   - Índice;
   - Introdução;
   - Desenvolvimento
     - Descrição e justificação das opções tomadas e dos métodos utilizados;
   - Resultados Experimentais;
   - Conclusões;
   - Bibliografia.
2. Justificar todas opções tomadas, nomeadamente, as transformações aplicadas e a definição dos elementos estruturantes (mas todas, não só estas)
3. Pode fazer-se em notebook Jupyter, mas deve estar organizado como um relatório
4. Nas imagens de teste podem aparecer moedas de 2euros
5. Entrega: 12 de novembro.

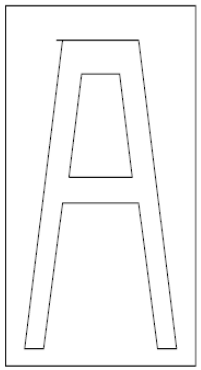
Quando as características básicas (área, centróide, eixos, buracos/lagos, baías, etc...) não são suficientes, é necessário uma representação com um nível mais alto de abstração.

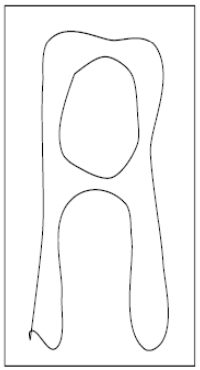
Representação por Grafo de Estrutura

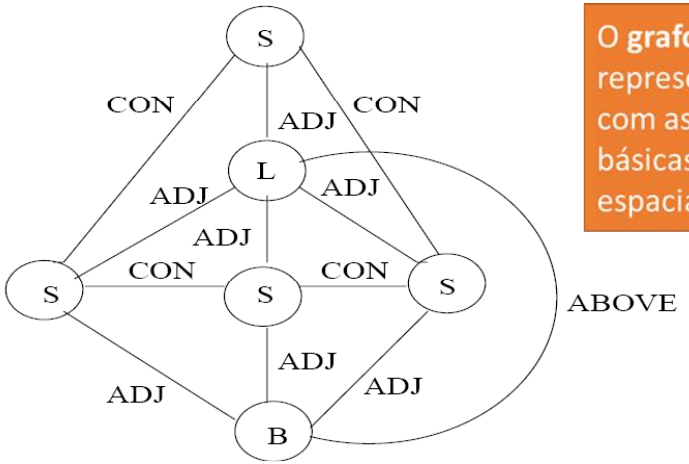
0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	1	1	0
0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	0	0	0
0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	1	1	0
0	1	0	0	0	1	1	1	0	0
0	0	1	1	1	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	0	0
0	1	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	1	0
0	0	0	1	1	1	1	1	1	0
0	0	1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0
0	0	1	1	1	0	0	0	1	0
0	0	0	0	1	1	1	1	0	0
0	0	0	0	0	0	0	0	0	0









O **grafo de estrutura** representa um objeto com as características básicas e suas relações espaciais

Existem métodos para relacionar estes grafos, mas podemos pensar que o grafo poderá ser transformado num vector de características e realizar uma classificação como as anteriores. Por exemplo, pode-se formar um vector com o número de vezes que ocorre um tipo de relação.

Outros exemplos de "Baías" e "Buracos/Lagos"

a) original b) bay and lake c) lid

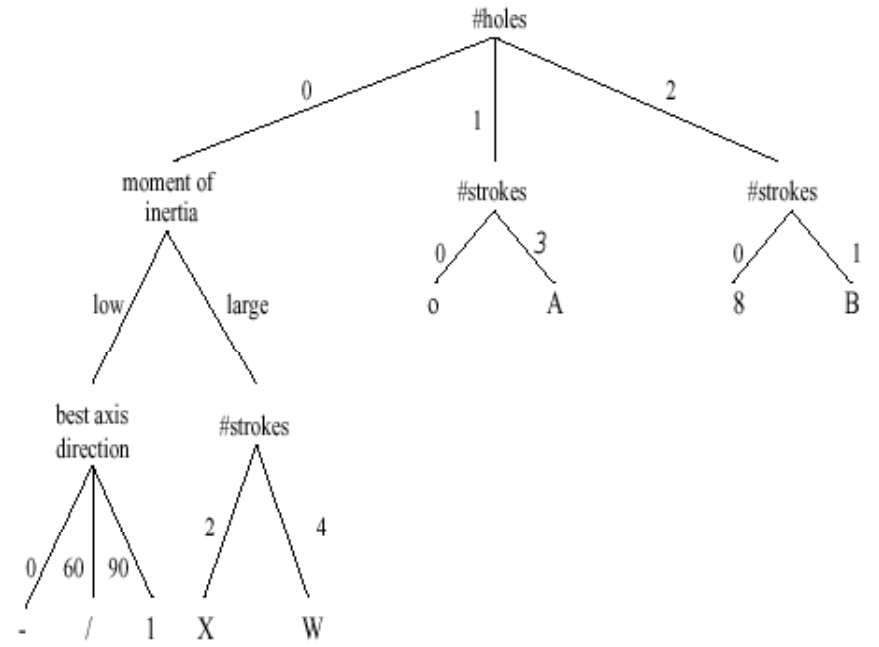
a) Um 6 manuscrito
b) A baía e o lago extraídos através de operadores morfológicos
c) O topo da baía extraído com subsequente processamento morfológico

Que características podemos tirar destes elementos?

Classificador - árvore de decisão

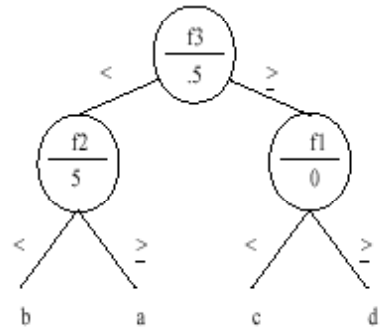
input: feature vector with [#holes, #strokes, moment of inertia]
output: class of character

```
case of #holes

   0: character is 1, W, X, *, -, or /

        case of moment about axis of least inertia

           low: character is  1, -, or /

                  case of best axis direction
                     0: character is  -
                    60: character is  /
                    90: character is  1

          large: character is W or X

                  case of #strokes
                     2: character is X
                     4: character is W

   1: character is A or 0

        case of #strokes
           0: character is o
           3: character is A

   2: character is B or 8

        case of #strokes
           0: character is 8
           1: character is B
```



f1	f2	f3	CLASS
3	6	0	a
-5	9	1	c
4	5	1	d
7	4	0	b
-1	10	0	a
2	6	1	d
-2	2	1	c
-1	3	0	b

Training Data

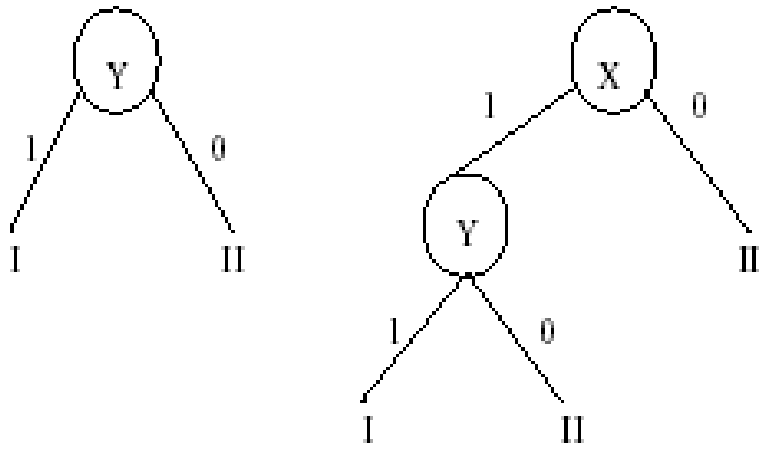


Decision Tree

Exemplo simples

X	Y	Z	Class C
1	1	1	I
1	1	0	I
0	0	1	II
1	0	0	II

Training Data



Two Possible Decision Trees

- Necessidade de automatizar o processo: problemas reais podem facilmente envolver dezenas ou mesmo centenas de *features*

- No caso das *features* tomarem valores não binários, como escolher os limiares de decisão?

Teoria de informação e as árvores de decisão

- A entropia de um conjunto de acontecimentos $x = \{x_1, x_2, \cdots, x_n\}$ é dada por

$$H(x) = -\sum_{i=1}^{n} p_i \log_2 p_i$$

onde p_i é a probabilidade do acontecimento x_i

- Dado um conjunto de *features* seleccionadas, quais são aquelas que são mais informativas para o processo de classificação?
- O conteúdo de informação que a *feature* F introduz, para a determinação da classe, C, é dada pela seguinte expressão

$$I(C;F) = \sum_{i=1}^{m}\sum_{j=1}^{d} P(C = c_i, F = f_j) \log_2 \frac{P(C = c_i, F = f_j)}{P(C = c_i)P(F = f_j)}$$

Relembrar Probabilidade Condicionada

↑

Informação mútua

P(C), P(F) e P(C|F) podem ser estimadas a partir das frequências relativas.

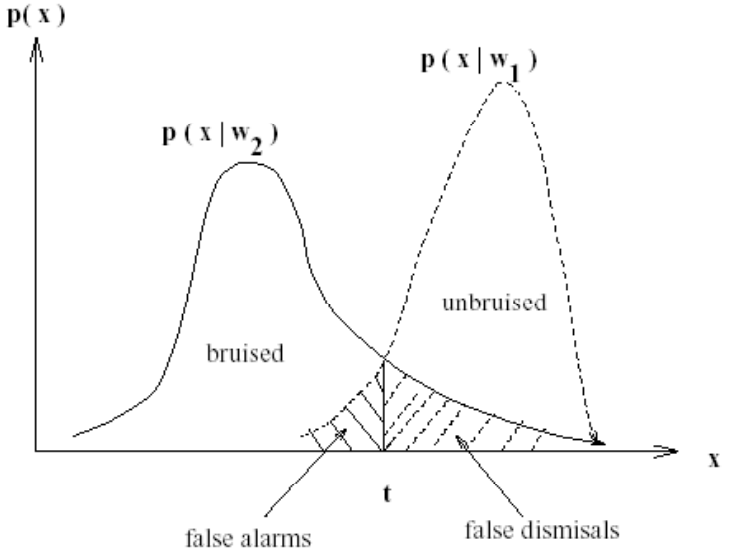
Classificador MAP

- Classificador MAP: um objecto é classificado na classe que é mais provável pertencer, após observação das suas *features*

$$P(w_i \mid x) = \frac{p(x \mid w_i)P(w_i)}{p(x)}$$

P(x): Relembrar lei da probabilidade total.

- Informação necessária:
  - Distribuição do vector de *features*, para cada classe $\longrightarrow$ $p(x \mid w_i)$
  - Probabilidades a priori de cada classe $\longrightarrow$ $P(w_i)$



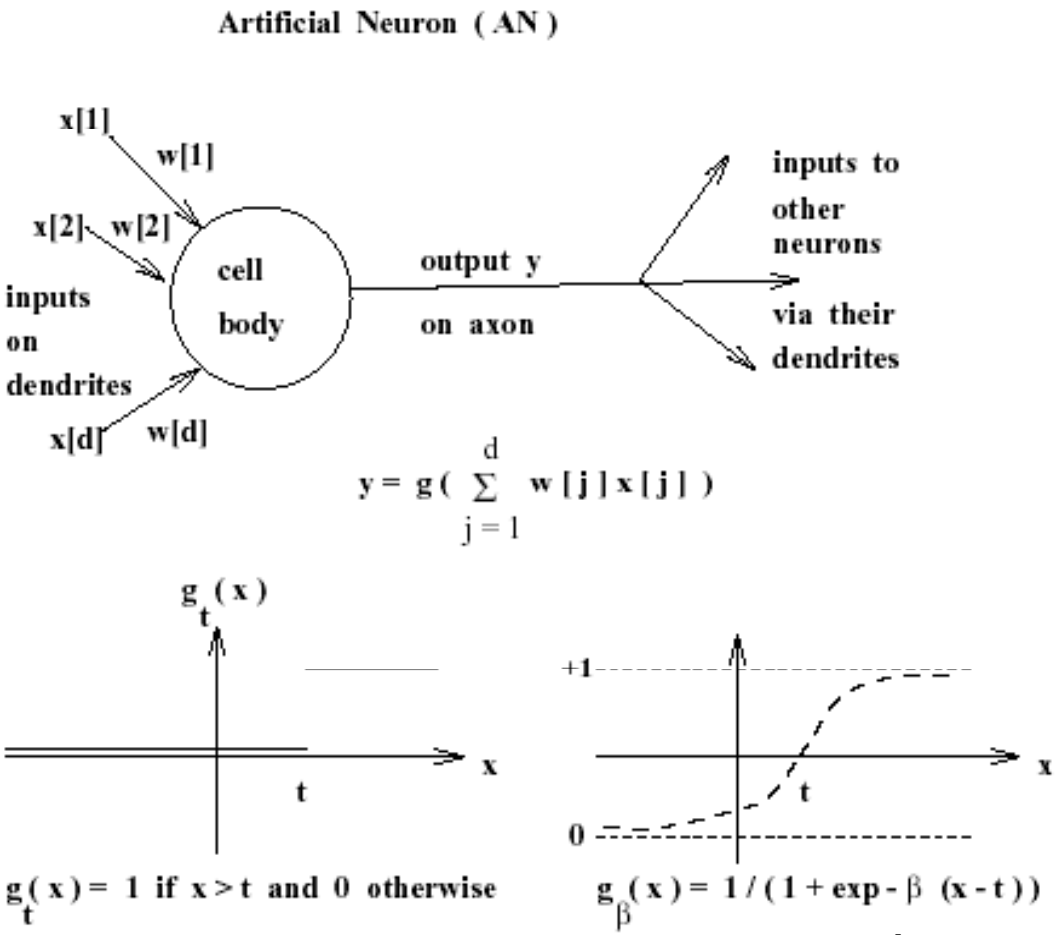
Como calcular estes parâmetros?

"A Neural Network is a massively parallel distirbuted processor made up of simple processing units, which has a natural propensy for storing experimental knowledge and making it available for use. It resembles the brain in two respects:

1. Knowledge is acquired by the network from its environment through a learning process
2. Interneuron connection strengths, known as synaptic weights, are used to store the acquired knowledge"

In Neural Networks (ch 1), Simon Haykin (1999)

Redes neuronais



Função Sigmóide

- Equação do hiperplano

$$\sum_{j=1}^{d} w_j x_j - t = 0$$

$$\sum_{j=0}^{d} w_j x_j = 0 \qquad w_0 = -t \quad x_0 = 1.0$$

- Espaço de features aumentado:

$$\tilde{x} = \begin{pmatrix} 1 & x \end{pmatrix}$$

- Entrada/saída em cada neurónio

$$y = g\left(\sum_{j=0}^{d} w_j x_j \right)$$

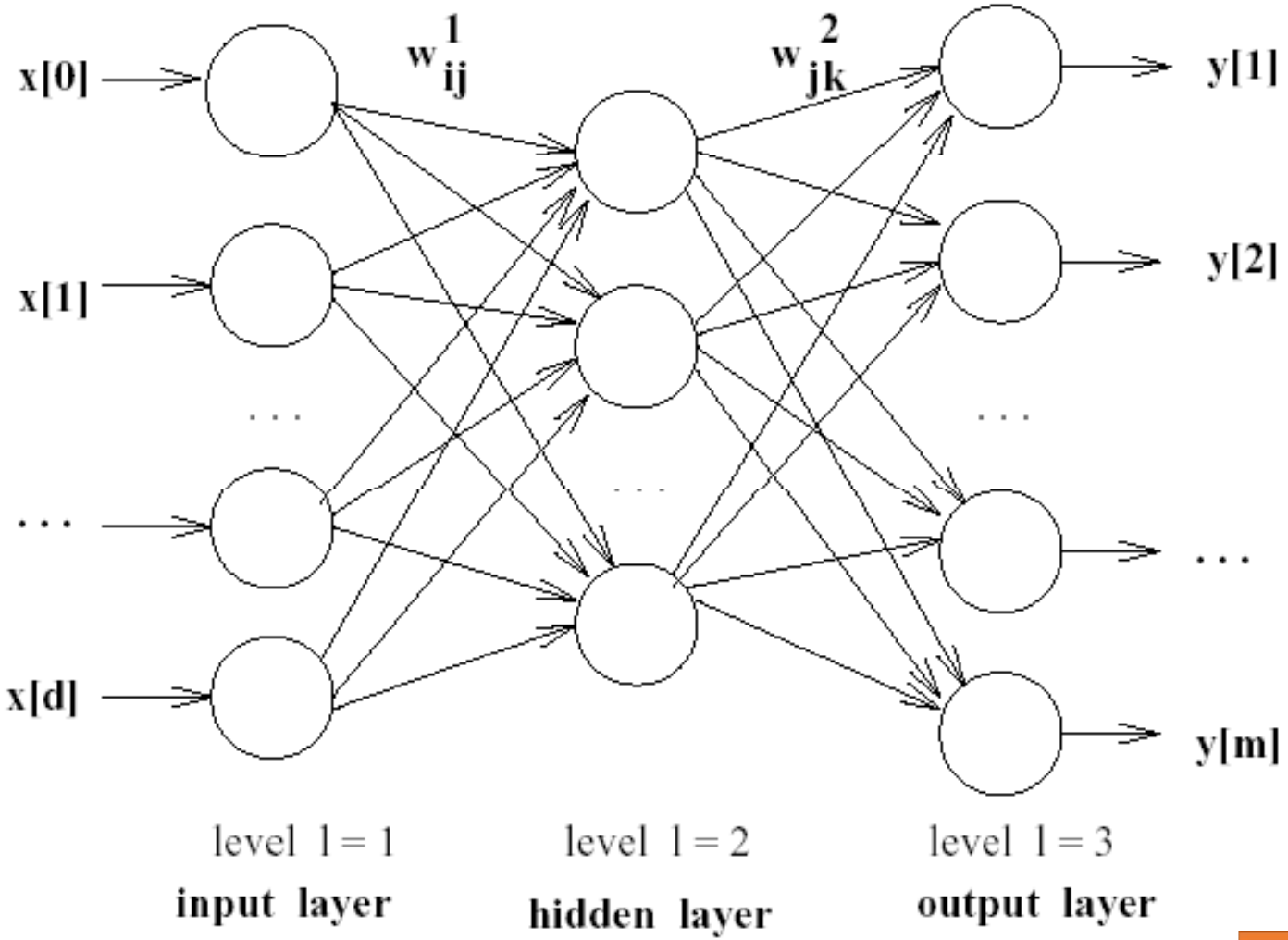
Discriminantes lineares – perceptrão de incremento simples

Aprender ←→ Adaptar pesos

Compute weight vector **w** to discriminate Class 1 from 2.
S1 and **S2** are sets of n samples each.
gain is a scale factor used to change w when x is misclassified.
max_passes is maximum number of passes through all training samples.

```
procedure Perceptron_Learning(gain, max_passes, S1, S2)
{
input sample sets S1 and S2;
choose weight vector w randomly;
"let NE be the total number of samples misclassified"
NE = check_samples ( S1, S2, w);
while ( NE > 0 and passes < max_passes )
   {
     training_pass ( S1, S2, w, gain);
     NE = check_samples ( S1, S2, w);
     gain = 0.5 * gain;
     passes = passes + 1;
   }
report number of errors NE and weight vector w;
}
   procedure training_pass ( S1, S2, w, gain);
{
   for i from 1 to size of Sk
   {
     "scalar, or dot, product ∘ implements AN computation"
     take next x from S1;
     if ( w ∘ x > 0 ) w = w − gain * x;
     take next x from S2;
     if ( w ∘ x < 0 ) w = w + gain * x;
   }
}
```

Rede *feedforward* multicamada

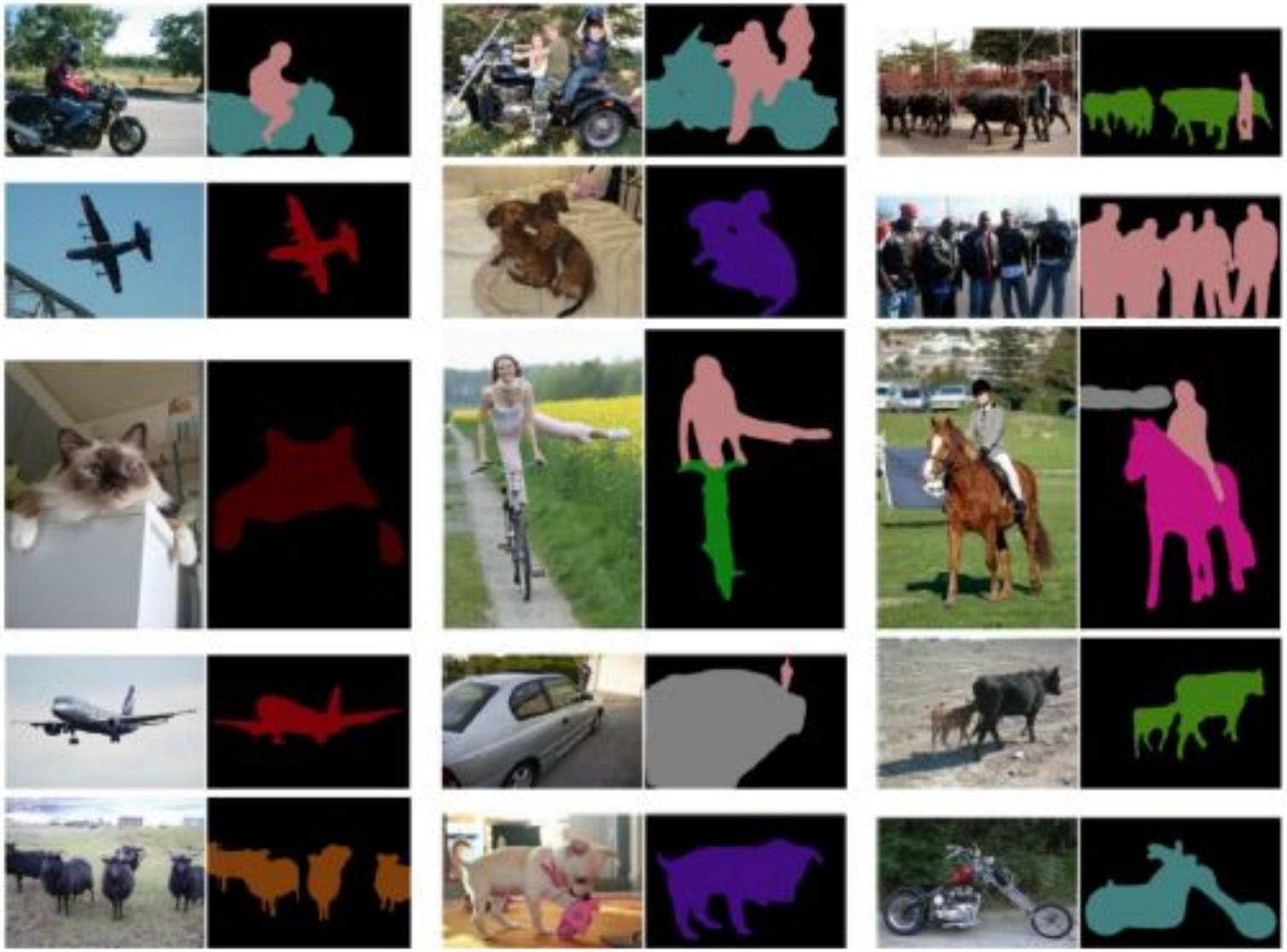


Aprendizagem → algoritmo de retropropagação

Backpropagation Ref: Neural Networks (ch 15.7), by Simon Haykin (1999)

Image segmentation using Deep Learning: a survey

https://arxiv.org/abs/2001.05566



Results from the work described in https://arxiv.org/abs/1706.05587